TPUG Newsletter

Views and News of Toronto Pet Users Group
c/o John Easton, 258 Lake Promenade, Etobicoke, Ontario, M8W 1B3 phone (416) 251-1511

Fall— 2011

From the President -

Fall is upon us and the summer is over, but what a successful summer it's been for TPUG! The fall and winter seasons promise to be just as exciting for TPUG as the spring. Quite a number of activities and interesting meetings are coming up in the next few months which will lead us as a club well into the year 2012, so stay tuned!

With the arrival of fall and winter comes the arrival of World of Commodore 2011. Yes, World of Commodore is upon us once again! This year's show will be held at the Admiral Inn in Mississauga. If that location sounds familiar, you're right. This is the same venue that the World of Commodore 2009 was held. This year's show promises more of the same (demos, vendors, freebie table) but more of the fun and surprises that make up our signature show so don't miss it! Conforming to WOC tradition, last year's show was the first Saturday in December. This year's show is also the first Saturday in December, just in time to pick up something special (and vintage) for the holidays!

And speaking of shows, this past summer TPUG participated in the annual FanExpo show, one of the largest shows of its kind in North America, attended by an average of 64,000 people during that weekend. This was TPUG's first time at this show and it was a success. Our booth was located in a room with the Personal Computer Museum and the Nintendo gaming club and we had a constant stream of people that visited our booth to try out our vintage Commodore computers. Shows like FanExpo help to get the word out that TPUG is still actively supporting the Commodore community. We hope to be involved again next year, as well as keeping our eyes and ears opened for other opportunities and shows. Stay tuned.

In order to beat away the fall and winter doldrums, I typically gravitate back towards the Commodore 64. Most folks I know enjoy the many games that were available for the C64. The number of games made specifically for the C64 is enormous and includes both commercial games as well as the many 'type in' games that were available in publications such as Compute's Gazette, Run, and Ahoy! I used to spend hours typing in the latest games from Gazette, saving them to the trusty 1541 (or cassette if I didn't mind waiting a while), and then excitedly typing 'run' to see the game in action! The games coded for the Commodore 64 were among the most interesting games made for any computer and provided hours of fun. I still enjoy popping in a disk (or cassette if I don't mind waiting a while) with some of my favorite games on it for more nostalgic fun.

But as many know, educational and productivity software for the Commodore 64 is my forte. The Commodore 64 had tons of software that fell under the categories of education, productivity and business. These were great utilities that helped to enhance the Commodore experience. While most games for the Commodore 64 have been archived and preserved, educational and productivity software have not been archived and preserved as much. Hopefully this will change in the future. In the meantime, I try to do my part ensuring this segment of the Commodore legacy lives on.

Commodore and other retailers put out database software for tracking inventory, payroll, records, you name it. One of these was called The Manager, a very handy database that was widely available. You could set up fields and define the type and length of the field, then enter your data and have The Manager keep track of and organize basically anything you wanted. Beyond The Manager, there were many utilities for keeping track of records that were even used by some libraries. I even have a program for the C64 that keeps track of your shopping purchases and let's you know when your cupboards are bare!

Several years ago I used Pocket Filer to keep track of my Commodore 64 programs that I have on disk and tape. Pocket Filer was a database utility, one of the earliest databases for the 64. It was a handy tool, I could do a lookup on any game by title and get all of the information about that game including my high scores and comments like 'wow, great game' and 'avoid this one!'. I initially started with a handful of games. I then added utilities and productivity programs to the database as well. I then

... continued on page 11 >>>

Member Information

Voice Info
We have discontinued our TPUG phone listing - contact members as listed here at home phones.

Website: **www.tpug.ca**
e-mail: **info@tpug.ca**

Membership Rates

$15 per year
else, with e-mailed Newsletter,
$10 per year

Board of Directors

President	Greg VanLaere
Vice President	John Easton
Secretary	Ernie Chorny
Membership Sec.	Ian Colquhoun
Treasurer	Ernie Chorny
Director	Leif Bloomquist
Director	Tom Luff
Director	Ian McIntosh
Director	Joe Palumbo
Director	Tom Williams

Librarians

Head Librarian	Greg VanLaer
Amiga	Ernie Chorny
C128	Tom Luff
CP/M	Ernie Chorny
GEOS	Joe Palumbo
C64	Joe Palumbo
C64 Education	Greg VanLaer
Comal	Ernie Chorny
Plus/4	Ernie Chorny
Vic 20	Ernie Chorny
PET/CBM/SuperPET	John Easton

Support

Mail	Tom Luff
Telephone	John Easton
Disk Orders	Librarians
Member Records	Ian Colquhoun
Meetings	Leif Bloomquist
and	Ernie Chorny
Shows	Tom Luff & Ernie Chorny
Webmaster	Leif Bloomquist

Newsletter

Editor John Easton (416) 251-1511
jeaston@rogers.com

Meeting Schedule

Westside and Amiga West: Third Thursday of the month (except summer) at Alderwood United Church, 44 Delma Drive. Delma Drive is just west of and parallel to Browns Line, south of the Queen Elizabeth Highway, north of Horner Avenue. From the west, exit QEW at Evans Avenue, east on Evans to next stoplight, south on Gair to Delma Drive. From the north or east, follow signs from QEW or Hwy. 427 to Browns Line, exit right to Evans Avenue, turn south on Gair (first stoplight) to Delma.

Contact - **Leif Bloomquist** (416) 737-2328 leif@schemafactor.com
or Ernie Chorny (905) 279-2730 chorny@tamcotec.com

From your Editor ...

Another year, another World of Commodore. You know, this event goes 'way back in the history of Commodore computers as we have grown to know and love them. I can well remember the days when Commodore hosted those fabulous events in the heyday of y'r own Personal Computer. Gazillions of vendors and representatives from the known universe would gather up there by the airport to dazzle and amaze us mere mortals as to the possibilities of life with a Commodore.

Well, now we're merely faced with keeping up with advances in the technology—still allowing us mere mortals to utilize our retro/ancient machines access to the world of to-day. Check out the articles about PetDisk and SUX 6400 in this issue. … We're not dead yet!

As an aid to this, you might like to check out our cartoon-of-the-month. Also, note the license plate of this car recently spotted at RIM. **SYS64738** is the code for a C=64 warm reboot! Must be hope for RIM yet ;-)

Joe Quittner has, as usual, provided us with helpful programs in the useful applications of Commodore technology, admittedly C=64 based, - an oddity your editor (PET-based) feels free to allow.

And, to explain an ancient reference to Jim Butterfield, back about page 10 — I recently passed on to our local secretary, Ernie, a bunch of old newsletters. However, in my haste, it appears that I left one issue of the Tri-City Commodore Computer Club on the floor. Thus the re-appearance of a Jim Butterfield article from 'way back. Ernie—I'm looking to you for the next re-prints.

TPUG Newsletter is meant to be published somewhat twice yearly by the Toronto Pet Users Group (TPUG). TPUG is a volunteer non-profit club dedicated to the service and support of owners and users of Commodore computers.

Articles, letters, tips, questions, art, etc. are welcome. Send hardcopy or disks "Attn: TPUG Newsletter", or use Internet e-mail.
Advertisements are also welcome. Member's small ads are free. Commercial ads are $100 per page with a $10 minimum.

Notice to new owners of SuperPet and CBM 8296 machines

TPUG has copies of the Waterloo LANGUAGE DISKS (3 in 4040 format) as supplied with the SuperPet on original purchase.

TPUG has the EXECUDESK disk (8050 format) as supplied with the CBM 8296 on original purchase.

These disks are an integral part of the operating systems of the above machines and since Commodore insisted on referring owners of these machines to TPUG for service, we have added these somewhat proprietory (and also virtually unobtainable) disks to our library - all part of the TPUG mandate of service to our members.

We also will attempt to search out copies of original program disks to replace corrupted disks. In this category you will find such programs as VISICALC, WordPro, and PaperClip.

TPUG Annual General Meeting

Take notice that the annual meeting of TPUG will be held in conjunction with the regular January 19th 2012 meeting, only earlier—7:00 p.m. This is your opportunity to discuss club issues and to contribute to the organization as desired.

Typical items covered at an Annual Meeting are, but not limited to: quorum, minutes of the previous meeting, reports on activities within the club, new business, nominations of officers and election of officers.

If you are unable to attend, and wish your proxy vote counted, please ensure that your assigned proxy is in the hands of the Board prior to the meeting. E-mail to the vice-president, jeaston@rogers.com will be considered a legitimate means of submitting a proxy to the Board.

Typical Proxy Voting Format

This will allow ... to vote on my behalf on all matters at the Annual Meeting of TPUG held 19 January 2012.

Name ..... TPUG Member # .....

Signature ..... (difficult via e-mail) . . .. Date .....

Reminder.. memberships cease at year-end .. renew NOW

TPUG Logo Contest

Well now … the Logo Contest hasn't really produced anything like the competition we expected from you avatars. Being so underwhelmed, we're not publishing those entries received so far (as promised in the Spring issue of the TPUG Newsletter) and have decided to extend the contest closing to noon on the day of World of Commodore '11, that is, noon of December 3rd.

For an overview of the "rules", see page 9

Joe's World ... 13 - Disk Drives

The Commodore 1541 disk Drive contains a small computer, dedicated to do certain tasks. The disk drive's Read-Only-Memory (ROM) contains about 16000 bytes, while the Random-Access-Memory (RAM) contains storage for about 2000 bytes.

Disks can contain program files (PRG), Sequential files (SEQ), User files (USR) and Relative files (REL). Also, Random files are available.

All disks contain a directory that can be loaded (load "*",8) and then listed, showing the length of each item, its name and also its type. The most often type used are Program (PRG) files. To read their contents, simply load the program and then type list or lI (lower case l and upper case I) followed by pressing the RETURN key.

Here us a program to read Sequential (SEQ) files:

```
0 rem "READ SEQ FILE BY JOE QUITTNER
10 print cH(144)cH(14)
100 goto 8000
997 :poke 204,0:get g$:if g$=""then 997
998 :if peek(207) then 998
999 :poke 204,1:return
4000 :rem"   Printer.  PRINTERS DIFFER, THIS
WORKS FOR THE ONE I USE.
4010 print"  Printer on? ";gosub 997:?g$
4012 p=0:if g$<>"y" then return
4020 p=1
4021 close5:close7:open 5,4,5:open 7,4,7
4030 for i=1 to 10
4032 pR5,cH(27)cH(108)cH(5):rem "Left margin
= 5
4034 pR5,cH(27)cH(81)cH(77):next:rem "Right
margin = 77
4039 print"  Set Interface to Junior":return
8000 :rem"  Read disk file
8002 gosub 4000:input"  FILENAME";f$:if p
then pR7,f$":"
8010 open 8,8,"0:"+f$+",s,r
8012 : get#8,h$:s=st:if p then pR7,h$;
8013 print h$;:if s=0 then 8012
8020 print:close 8:if p then pR7
8021 end
```

Think of the old phonograph music disks; they have a single track like a spiral and the information on that track starts on the outside, with the needle reading the music from there continuously until the end of the spiral track is reached near the centre of the disk.

The disks used by the 64 and C64 computers have no spiral track, but instead there are 35 selectable tracks in concentric circles on the disk. Each of these tracks is subdivided into selectable blocks, also called sectors, each carrying 256 bytes, with the outside tracks 1 to 17 having 21 blocks/sectors each, tracks 18 to 24 have 19 each, 25 to 30 have 18 each and the near-centre tracks 31 to 35 have 17 each on them.

The big advantage of using disks over using tapes is that on disks any block/sector on any track can be selected, and even any of the 256 bytes on any block/sector can be pointed to, read, and written into by the disk drive.

Programs (PRG) are taken from the disk by the load command, and are written onto the disk by the save command; load"filename",8 for example will take the program called filename from the disk and load it usually into memory starting at 2049, with a compulsory zero in 2048.

There is an option to load the PRG elsewhere by using load "filename",8,1 if that program, located elsewhere in memory, has been saved previously by the save "filename",8,1 command. The number 8 indicates a specific disk drive; more than one disk drive can be in use, their device numbers being 8 to 11.

To change, for example, the device number of a disk drive from the default 8 to 9, use this, or a similar, program:

```
10 open 15,8,15:rem" Tells drive 8 that a
change is coming
12 pR 15,"m-w" cH(119)cH(0)cH(2)cH(9+32)
cH(9+64:rem "Details of the required change
```

Instead of this software change, a hardware change of the device number can be made to a disk drive by removing one or two specific links on the printed circuit board inside the disk drive.

To open a SEQ file for reading, often containing the data used by a PRG file, use for example.

```
open 2,8,2,"0:filename,s,r"
```

where the first 2 is the optional number selected by the user, the 8 is the device number of the disk drive, the second 2 is the number of the selected buffer (a portion of the memory located inside the disk drive), the 0 indicates that more than two buffers can be used, followed by the name of the SEQ file. The **s** indicates that it is a SEQ file, and the **r** will cause reading from that file. Instead of the r, w can be used if, instead of reading from, writing into that SEQ file is to be done. Note that each disk can be write-protected by covering the slot on the side of the disk, and if a file name exists already there will be a warning message, followed by a stop. These can be disabled by using "@0: instead of "0 above.

Instead of the s, p will refer to a PRG file, or u to a User (USR) file, mainly used for machine language.

Closing the files is simple; close 2 will close file 2. After completion of a saving operation, or otherwise after completion of writing into a disk, the file MUST be closed immediately.

For random files, the open statement includes the optional file number, the device number, the number of the channel and "#" indicating to the disk drive that any buffer can be used by the disk drive. For example:

```
open 2,8,2,"#"
```

Several commands are available for random files:

To read a block of bytes use b-r.
To write into a block of bytes, use b-w.
To find (allocate) a still free block and then reserve it, use b-a.
To free (make available) a reserved block for a possible new use, use b-f.

To get the pointer to move to a specific byte inside a buffer, use b-p, so that this specific byte can then be either read from, or written into.

The user1 ("u1" or "ua") command will enable the reading, using input# or get# to follow, of a whole block, not only just one byte as per pointer position.

To read a whole block from a disk, use:
`print#file#,"u1:",channel;drive;track;block.`

For writing into a whole block, using print# to follow, use user2 ("u2:" or "ub:") instead.

Joe Quittner, TPUG, June 19, 2008

Note ... Joe's World 14 was an index of articles to that date, we would hope to utilize this with a complete index at the end of his current series ... at the moment, numbering 31. More to follow.

Joe's World 15 ...
Tracing a Basic Program's Progress

If a program doesn't work, or doesn't work as expected, then it helps to have some way to identify where the trouble starts. This program is activated by typing in direct mode (without line numbers) sys 680. Then line numbers will be printed out without interfering with whatever the target programs would print out, if anything, normally. Typing sys 683 will deactivate this program. It makes use of one of the vectors specifically designed to permit users to interpose a machine-language program, such as this one, into a basic language program.

A vector is a pair of memory locations, here 776/777, that BASIC makes available as entry point of a user's machine language program, if used, and normally points to the memory location for BASIC to continue. If a user program is interposed, then it's exit address must be that normal continuation address. In this dase 776/777 points normally to 228/167 (=42980), a part of the routine that executes the program.

```
0 rem "trace BASIC" by Joe Quittner
10 print cH(14)cH(144)
12 print "   For TRACE ON type sys680
14 print "   For TRACE OFF type sys683
1000 for i=680 to 767:read a: if a>255 then gosub 4000:
poke 195,0: poke 196,0: new
1010 poke i,a:next:stop
1020 data 56,176,11: rem "Trace on
1021 data 169,228, 141,8,3, 169,167, 141,9,3, 96: rem
"Trace off
1022 :data 169,196, 141,8,3, 169,2, 141,9,3, 96
1024 :data 169,19, 162,3, 160,255, 32,186,255, 32,192,255:
rem open file 19
1026 data 162,19, 32,201,255: rem from now on print into
file 19 (screen)
1027 data 165,57, 197,195, 208,6: rem same?
1028 data 165,58, 197,196, 240,20: rem same?
1032 :data 165,57, 133,195, 165,58, 133,196: rem save
new line number
1033 data 166,57, 165,58, 32,205,189, 169,32, 32,210,255:
rem print it plus space
1034 :data 169,19, 32,195,255: rem close file 19
1036 data 76,228,167, 999: rem "Continue normal BASIC
operation
```

Lines 1000 and 1010: This program is loaded into 680 to 764, and then cancels itself, so that the target program, with its trouble, can then be loaded. Also 195/196 is initialized to zero; it will be used to save the most recent line number.

Line 1020, Trace on: 680 sec (set carry), 681 bcs (branch if carry is set) to 694: The carry flag is set and because it is set, the interposed program will continue at 694.

Line 1021, Trace off: 683 lda loads .A, the accumulator (780), with the number 228 and stores it into memory location 776, the vector. Similarly number 167 is stored into 777, so that now the vector points again to BASIC's normal continuation location, 65484, no longer to the interposed program. See also line 1036.

Line 1022,: The starting address, 705, is loaded into the vector if the "Trace on" has been activated.

Line 1024: 705: The optional file number, 19, is loaded into .A; I have chosen 19 because it is unlikely that any of the target programs will use file 19. 707: The output device number, 3, is loaded into .X (781); 3 is the screen, use 4 if you want the printer instead. 709: If there would be a secondary address, it would be loaded into .Y, but there is no secondary address for the screen, therefore 255 is loaded into .Y (782). If instead of the screen you use the printer as the output device, then use 7 (character table for capital plus lower case letters) instead of 255. These three items having been loaded as precondition, they are fed into 65466, the pointer to the subroutine SETLFS starting at 65024 which only puts .A into 184, .X into 186 and .Y into 185. Next, 714 activates subroutine 65472 which opens the file described by 65466.

Line 1026: 717 loads .X with 19 and 719 goes to 65481 where the pointer (CHKOUT) leads to 62032, where the subroutine starts that takes from .X the file number (here 19) to be used from now on for output (here the screen).

Lines 1027-1028: Often in programs "loops" are used, when the program waits for the user to press a key, for example, with BASIC at high speed using the same line again and again. To prevent the trace printing a line number again and again, from 722 to 732 a comparison is made between the current line number and the previous line number, stored in 57/58. If they are the same, then the trace program is sent to its exit, at 754.

Line 1032: If this is a different line number, then 734-740, it is stored in 195/196, replacing the previous line number there.

Line 1033: 742-746: The current line number, located in 57/58 is fed into the subroutine starting at 48589, which, with the help of the subroutine starting at 58605, causes this number to be printed to the output device, here the screen. 749-753: Also a space is then printed with the pointer at 65490 (CHROUT) using the subroutine starting at 61898.

Line 1036: This is the machine language interposed program's exit. First, 754-758, file 19 is closed by the subroutine starting at 62097 which is pointed to by the pointer (CLOSE) at 65475. Next the program is led to the same address, 42980, that the vector 776/777 pointed to originally.

Vectors Listed

The vector at 768/769 points to the error subroutine that starts at 58251.

The vector at 770/771 points to the subroutine that starts at 42115. It operates the main loop of BASIC. Statements are either executed, or stored in memory as lines of the program.

The vector at 772/773 points to the subroutine that starts at 42364. It changes keywords in text into token numbers (128-203). For examples, end=128, for=129, next=130, ... go=203.

The vector at 774/775 points to the subroutine that starts at 42778. It produces text in English from a given token number.

The vector at 776/777 executes the task assigned to a given token number using the subroutine starting at 42980.

The vector at 778/779 points to the subroutine that starts at 44678. It works out the result of mathematical expressions and of mathematical functions.

To access any of these vectors directly, use the opcode 108, the indirect jump. For example, 108 8 3 uses the vector at 776/777 to jump indirectly to 42980.

By Joe Quittner, TPUG, October 16, 2008

Joe's World 16 ... Plotting on Screen - 40 x 25

This program for each given data point puts an optional symbol (most keys) on the screen in an optional colour, thus permitting several groups of data to be shown together. After the end of the plot, the user can write on the screen and/or manually add more points, such as to roughly connect these points if desired. To change the colour of what is being typed on the screen hold down the CTRL key while typing one of the number keys, or hold down the C= key while typing one of the number keys.

I have set a limit of 1000 data points (x-y) on line 10, but subject to memory space availability this can be increased.

```
0 rem"  Screen plot 40 by 25 by Joe Quittner
10 dim xy(1000,3)):rem 4 per point
100 print cH(144)cH(14)"          MENU:
101 print "a.Type into array        1000
102 print "b.Save on disk           3000
103 print "c.Load from disk         8000
104 print "d.Read data              2000
105 print "e.Plot on screen         7000
106 print "f.Type additional data   6000
190 print "     WHICH? ";:gosub 997:print g$
201 if g$="a" then 1000
202 if g$="b" then 3000
203 if g$="c" then 8000
204 if g$="d" then 2000
```

```
205 if g$="e" then 7000
206 if g$="f" then 6000
900 :print  "CHANGE?";
901 :poke 204,0: get g$:if g$=""then 901
902 if g$=cH(133) then gosub 1010: goto 900:
rem F1
903 if g$=cH(134) then gosub 1012: goto 900:
rem F3
904 goto 998
997 :poke 204,0:get g$:if g$="" then 997
998 :if peek(207) then 998
999 poke 204,1:return
1000 :rem "   TYPE INPUT DATA
1001 n=0:n1=0
1002 :input "Plotting symbol"; s$: s=asc(s$)
1003 input "Plotting colour number (0-15)"; c
:if c<0 or c>15 then 1003
1006 print "To change plotting symbol: press
f1 after CHANGE?
1008 print "To change plotting colour number:
press f3 after CHANGE?
1009 print "To end x,y input type 999,:
goto 1020
1010 :input "new symbol";s$:s=asc(s$):return
1012 :input "new colour number"; c: return
1020 :gosub 900:n=n+1:print n;
1022 input "x,y or 999,";x,y:if x=999
then 100
1030 xy(n,0)=x: xy(n,1)=y: xy(n,2)=s:
xy(n,3)=c: n1=n: goto 1020
2000 rem"    LIST DATA
2010 for i=1 to n1
2012 print i;xy(i,0)xy(i,1)xy(i,2)xy(i,3)
2014 next:gosub 997:goto 100
3000 rem"    SAVE DATA ON DISK
3002 input "OUTPUT FILE NAME";f$
3004 close 8:open 8,8,8,"@0:":+f$+",s,w":
print "Please wait!
3005 for i=1 to n1:print#8,stR(xy(i,0))
3006 print#8,stR(xy(i,1))
3007 print#8,stR(xy(i,2))
3008 print#8,stR(xy(i,3)):next
3010 close 8:goto 100
6000 rem"    TYPE ADDITIONAL DATA
6002 goto 1002
7000 rem"    PLOT ON SCREEN
7010 xn=1e30:xx=-1e30:yn=1e30:yx=-1e30
7012 for i=1 to n1:x=xy(i,0):y=xy(i,1)
7014 if x<xn then xn=x: rem min x
7016 if x>xx then xx=x: rem max x
7022 if y<yn then yn=y: rem min y
7023 if y>yx then yx=y: rem max y
7024 :next: xd=xx-xn:yd=yx-yn
7030 xk=39/xd: yk=24/yd: for i=1 to n1:
x=xy(i,0):xy(i,0)=(x-xn)*xk
7031 y=xy(i,1): xy(i,1)=(y-yn)*yk: next
7040 def fnr0(i)=int(i+.5)
7041 for i=1 to n: xy(i,0)=fnr0 (xy(i,0)):
xy(i,1)=fnr0 (xy(i,1)): next
7050 print cH(147): for i=1 to n1:
x=xy(i,0):y=xy(i,1):s=xy(i,2):c=xy(i,3)
7051 z=1984-40*y+x: cc=z+54272: poke cc,c:
poke z,s: next: end
8000 rem"    LOAD DATA FROM DISK
8001 input "INPUT FILENAME"; f$: n=0
8002 close 8: open 8,8,8,"0:"+f$+",s,r":
print "Please wait!
8003 :n=n+1:
input#8,xy(n,0),xy(n,1),xy(n,2),xy(n,3)
8008 if st=0 then 8003
8010 n1=n: close 8: goto 100
```

Line 10 dimensions (reserves memory space) of array xy, defines the number of data points (1000 maximum, optional), and defines the number of descriptors; there are 4 (zero counts as one): x, y, symbol s, and colour number c. The colour numbers are: black=0, white=1, red=2, cyan=3, purple=4, green=5, blue=6, yellow=7, orange=8, brown=9, light red=10, dark grey=11, medium grey=12, light green=13, light blue=14 and light grey=15.

The background colour of the screen can be changed by poking a colour number into 53281, and the colour of the screen border can be changed by poking a colour number into 53280.

Lines 100 to 205 are parts of the menu; press either a,b,c,d or e to start whatever is listed on the menu after that letter.

Lines 900 to 999 are for waiting. The cursor keeps blinking while the program waits for a key to be pressed by the user.

Lines 1000 to 1030 permit entries of x and y for each data point, and also, changes can be made to the symbol and colour on screen for use later while plotting. Data can be entered in any order, and even different data sets, in optional different colours and/or symbols can be entered in any order.

The data can be listed by lines 2000 to 2012. Any changes can be made by changing the contents of the xy array. For example, if in data point 4 the value of x should be changed to 6.5 then type in direct mode (no line number) xy(4,0)=6.5 0=x, 1=y, 2=symbol, and 3=colour number of data point 4 in this example.

Lines 3000 to 3010 permit saving the data on a disk. The printing of strings saves disk space; one space per item.

The saved data can be loaded from disk into the xy array using lines 8000 to 8010.

The minima and maxima are found by lines 7000 to 7024, and these values are then used in the calculation where on the screen the data points are to be plotted, and when found, the symbol is put there in the specified colour. 0,0 is the left bottom corner, and 39,24 is the top right corner of the screen.

By Joe Quittner, TPUG, November 20, 2008

PETdisk

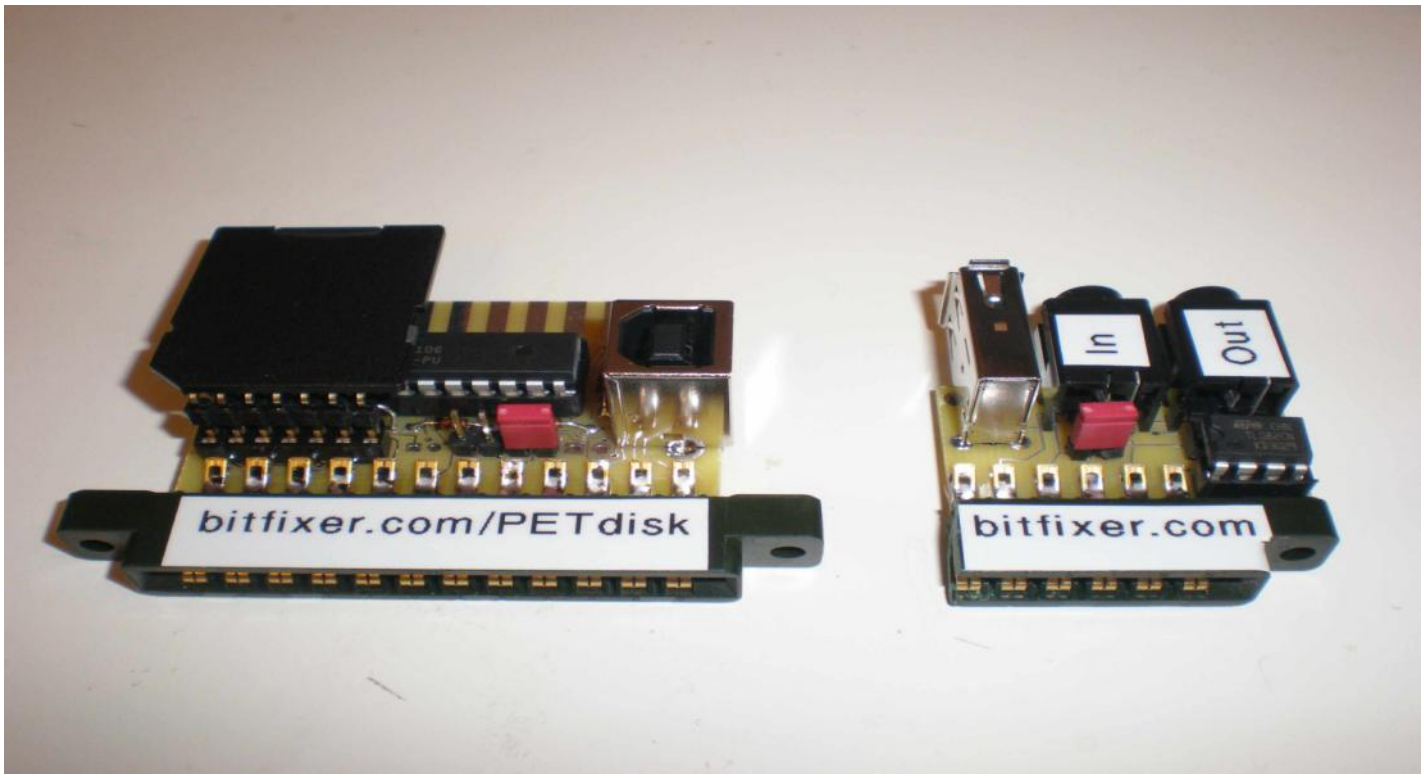
A review by Steve Gray

The wait is finally over! The first modern card storage solution for the PET and CBM-II machines has arrived. Its called the **PETdisk** and is produced by **Michael Hill** (aka Bitfixer). Kits are available for purchase at ***www.bitfixer.com*** for $25. Soldering is required, but if you're not comfortable doing it yourself please contact Michael for pricing of completed units. This nifty little device plugs directly into the IEEE port on the back of your machine and gives you a microSD card slot (microSD card not supplied). One of the great features of this unit is that it does not require an expensive and hard-to-find IEEE cable (*ed note* … check out the TPUG store). Also included is a small board that plugs into your cassette port to provide power to the PETdisk. Alternately, you can power the PETdisk using a standard USB charger/adapter or even your PC. As a bonus, the cassette board also provides audio in and out jacks so you can use a standard cassette player rather than the Commodore datasette. Or, you could hook the audio directly to your PC to store and playback virtual tapes.

Operation was simple. The PETdisk comes set as device 9 but can be jumpered as device 8, 9,10 or 11 as needed. Plug the PETdisk into the IEEE port, and the cassette board into the cassette port. Connect them using a standard USB cable (not supplied). Next, format your microSD card using FAT32 and copy your PET files onto it. Insert the microSD card into the PETdisk slot and you are done.

When all is working the PETdisk gives you basic storage that appears like a large floppy or hard drive. I was able to save over 300 PET ".PRG" files onto a card. Typing LOAD"$",9 loads in the directory. Use the LIST command to see the files, then LOAD "filename",9 loads the desired file. Note that BASIC4 commands currently are not supported, so you cannot use CATALOG, DLOAD or DSAVE at the moment. I did notice that the current firmware has trouble reading certain files. Michael has promised to update the PETdisk firmware to fix bugs and add support for BASIC4 commands in the future. Also, be aware that this device does not emulate any particular Commodore drive, it just works using a subset of generic Commodore DOS commands. I did not test the audio in/out features.

The PETdisk was first off the mark, but other solutions are on the horizon. The "PETSD" is also available now, and the "CBM4041" should be available soon. These promise to provide even more capabilities, but at additional cost. However, if you are looking for an inexpensive card solution that is easy to use, the PETdisk is for you!



SUX 6400 Audio Digitizer Cartridge

On July 24, 2011, at the Commodore Vegas Expo v7, the Sound Ultimate Xpander 6400 (SUX 6400) and its Digimaster 64 software made their premiere. Developed after hours and hours of work, the SUX 6400 is a new audio digitizer cartridge produced by the Fresno Commodore User Group and PDXCUG.org. From Chris Brenner of Autumn Technologies, Digimaster 64 is a re-release of his audio-digitizing and editing program.

The SUX 6400 with Digimaster 64 make up the most sophisticated audio digitizing hardware and software ever made for the Commodore 64 and the Commodore 128 (in C64 mode). With the SUX 6400 and Digimaster 64, you can capture audio into the Commodore and play back the recording - all in 8-bit sound quality. You can use the recording in your own programs. Full documentation is included.

The SUX 6400 / Digimaster 64 package has the following:
1. the SUX 6400 audio digitizer
2. a "flippy", two-sided disk with Digimaster 64 software
3. the Digimaster CD disc
4. a quick-start sheet of instructions

Below is Chris Brenner's original description of Digimaster 64.

Digimaster is a unique software product which allows you to process digital audio on your Commodore 64.

This advanced software utilizes a fully graphical interface with pull-down menus. Incorporated into its operation are many powerful features, such as Cut, Copy, and Paste, which makes editing sound as easy as editing text in a word processor. Using an optional audio digitizer [like the Sound Ultimate Xpander 6400], live sound can be grabbed into the computer, edited, and then saved to disk. There is even a utility included which will convert Amiga sound samples, giving you access to a vast library of sounds.

Probably the most impressive feature of this software is its ability to replay sounds in true eight-bit digital audio on your Commodore 64 without the need for extra hardware. This is made possible by a revolutionary method of controlling the sound chip inside the Commodore 64. The result is crystal clear audio reproduction.

Digimaster Features:
Fully graphical interface
Many powerful editing functions
Ability to replay sounds in true eight-bit digital audio without the need for extra hardware
Included software which allows replaying sounds from your own programs

Requirements:
Commodore 64 computer
Commodore 1541 or compatible disk drive
Joystick or mouse (mouse highly recommended)

Photos of the SUX 6400 and its production can be found throughout *http://retro-link.blogspot.com*

The SUX 6400 / Digimaster 64 package is priced at $22 without a cartridge case or $25 with a case. Within the U.S., shipping is via U.S. Priority Mail flat rate for $5.20. For other countries, shipping rates are to be determined. Send a check or money order to the

FCUG
502 Whitney Lane
Visalia, California 93277-1940
U.S.A.

Payment by PayPal is available; send an e-mail for more information on that payment option.

Sincerely,
Robert Bernardo
Fresno Commodore User Group
http://videocam.net.au/fcug

TPUG LOGO contest rules:

1. Logo can be black and white or colour.

2. Design must be easily converted to black and white or monochrome.

3. No tagging of artwork.

4. Multiple submissions are allowed.

5. Any format accepted, drawn or computer rendered. Computer rendered artwork should be in an easily readable file format (TIFF, JPEG, etc.).

6. Should be usable on letterheads, business cards, newsletters and signs.

7. Contest is ***open to all*** and voting is restricted to paid up TPUG members as of the date of the contest close.

8. All submissions become property of TPUG.

9. The TPUG board reserves the right to break a tie in the voting.

Send entries to our mailing address, else, via e-mail to *contest@tpug.ca*

Winners will be announced at a later date

Prizes will be awarded for the best design.

1st prize: $100 Gift Certificate for Jim Brain's store Retro Innovations
2nd prize: TPUG library CD
3rd prize: One year TPUG membership

It Just Happened ...

Jim Butterfield

from Tri-City Commodore Computer Club ... May 1994

as reproduced from the August 1992 Commodore Users of Bartlesville (CUB) newsletter. (Harold McCollum, 1920 Santa Fe, Bartlesville, OK 74003)

I'm often asked about two features of the Commodore Machines. The first is the non-standard code, Commodore ASCII. Why did Commodore choose it? The second is the **STOP** key. Why doesn't it work during user input? Oddly enough, the answer to these two is related, and it is intertwined with the way these personal computers grew in the early days.

Back in the renaissance days of 1975 and 1976, personal computers were mostly home brew. You'd buy a mess of chips and would spend long hours attaching them to a circuit board. Sometimes you'd design your own system and sometimes you'd buy a kit. Even with a kit, you'd customize your machine, often based on what was available and cheap in your neighbourhood. Some hackers soldered their connections, others used an almost-forgotten construction technique called "wire-wrap". Back in those early days, each computer was unique, reflecting the style, pocketbook and construction skills of its owner.

One of the biggest problems - technical, financial and practical - was how to attach input/output to the computer. Keyboards could be found, although they were generally a collection of "uncoded" switches, so that the hobbyist had to figure out hardware and software methods to connect them. Output was more of a problem. Video displays were not common, and the logic and circuitry needed to allow character display on a CRT was not standardized. Many of the early "kludge" displays had barely enough logic to display a range of 64 characters. This would allow upper case alphabetics, numbers and punctuation, but no lower case.

"Rich" users would find a way to obtain a teleprinter. Suitable scrounging might turn up a Teletype (TM) model 32 or 35, which used ASCII code, upper case only. These terminals were also used by business for "time-sharing" where a number of users submitted their work to a central computer from their terminals. Again, these devices had no lower case, only capitals, and they shaped our concept of the nature of a computer terminal.

Now comes the first Commodore machine, the PET 2001, with its tiny keyboard and built-in cassette deck. The only alphabetic characters that seemed to be on these machines were upper case (capital letters), a carryover from teleprinter machines and early BASIC concepts. The competing machines of that generation - the Apple][and the TRS-80 model 1 - had only upper case letters.

In fact, the Commodore was ahead of its competition in that it did have lower case in the first machine. These lower case characters were not visible unless you knew to **POKE 59468,14**, at which time many of the shifted graphics characters would change to lower case alphabetics. Thus, the heart graphic would change to lower case (not upper case) S.

Now, at that time, the character set used by Commodore was mostly true ASCII. An upper case A, for example, was code 64 - correct ASCII. This is still true of Commodore machines in graphics mode.

But users started to get into word processing and it didn't seem natural to use the SHIFT key to get lower case. You expect to press the SHIFT key for upper case. In their next model, Commodore conceded the point by flipping upper case with lower - in text mode only of course. As a consequence, compatibility with ASCII was greatly reduced.

That's how we lost it, and that's why we need to translate characters when sending to a modem or to a non-Commodore printer. Commodore was more or less pushed into it during the evolution of their machines. Did they have any alternatives? Yes, but none of them seemed good. They could have completely switched around the character set, graphics and all, but we would have lost compatibility with early machines. They could have converted the graphics set so that in the graphics mode, you would have only lower case alphabetics, but that would look terrible on graphic screens. They could have ended up with the clumsy system on other computers whereby you cannot write a program unless you put the SHIFT Lock down.

The important point is this: Commodore was not engaged in some sneaky plot to invent their own character set. They just ended up being maneuvered into that position.

What about the **STOP** key. Why doesn't it work during a user input? Again, it goes back to the teleprinter machines of the early computers. The pioneer microcomputers were often hooked up to a teleprinter or other serial device. ("Serial", here means something like the RS-232 interface - not the Commodore serial bus.) They were so constructed and programmed that they could do only one thing at a time. They could choose one of the following:

(a) watch the keyboard line for incoming characters.
(b) do other computing.

They could not do both. When Microsoft (TM) Basic was written for these early computers, it was known that BASIC must "freeze" and give up control of the system in order to get input. In other words, once you typed **RUN** on your teleprinter and pressed **RETURN,** your keyboard was dead until the computer decided to come back and

look at it again. BASIC would not look at the keyboard unless the program stopped or an input was required.

The **STOP** key was often on the computer itself, not on the keyboard. The program to check this key was implemented as part of BASIC. As your BASIC program was running, the interpreter would check the **STOP** key at frequent intervals and stop the program if the key was found pressed.

When BASIC executed an INPUT statement, it would suspend operation. The computer needed all the time available to watch the keyboard, and BASIC execution would be "frozen" until the desired input had been received. Naturally, BASIC wouldn't be checking the STOP key during this time.

In early days, only the Commodore computer left the keyboard "alive" while BASIC was running. It did that with a clever system, still used, called "interrupts", which allows BASIC and the keyboard to run virtually at the same time. So - on Commodore machines only - it would have been possible to watch the **STOP** key, even during input. But Commodore purchased their BASIC , which had been written for the "standard" machines of the day. So, BASIC didn't know of this advanced feature, and the **STOP** key did not work during input.

Funny, isn't it? It's as if the early Commodore machines with built-in lower case and interrupt keyboard servicing were so far ahead of their time that they were hampered by these enhanced features.

>>>> From the President ... continued from page 1

tried to sort the database in alphabetical order. Unfortunately, this took over an hour to sort! Eventually I split the database it into two parts: Games and Utilities. I still use Pocket Filer to keep track of my games, and sorting is quite a bit faster now. My goal is to try sometime in the near future to migrate to faster database software such as The Manager. I've just got to find some time to do it. Oh well, stay tuned.

We have recently completed an inventory of our storage locker, so a new listing will soon be published on our webpage in the TPUG Store section. Stay tuned for that as well. Fall is not usually associated with cleaning and tidying up, but we typically pick fall to do our locker inventory so we know what items we have on hand at World of Commodore, plus it helps to keep our webpage listing fresh. Keeping track of our inventory also helps us to find items quicker. Several years ago we started to use an inventory system that tracks items in the locker by item in a box, with the box having a specific number. This accounts for the codes that you see in the store listing. For example, 'SW-24' denotes 'software, box 24'. When we go to find the item in the locker, we know to look for box number 24. It beats the old way, where we simply opened up the box, looked inside and if it wasn't in the box, it was on to the next box!

Whatever activities you have planned for the next few months make sure they include the Commodore. Don't forget about our December meeting (holidays and vintage Commodore, an annual tradition), and don't forget about our logo contest (details in the newsletter), and definitely don't forget World of Commodore.

There's a lot in store for the next few months, so stay tuned to TPUG!

Happy Commodoring!
Greg Van Laere
TPUG President

The World of Commodore

December 3, 2011 from 10AM

At the Admiral Inn which is located on the north side of the QEW, just west of Erin Mills Parkway Mississauga, Ontario, Canada

Come see demos, vendors and guest speakers regarding new and innovative products for Commodore computers.

Visit us at: www.worldofcommodore.ca

TORONTO PET USERS GROUP
C/O JOHN EASTON
258 LAKE PROMENADE
ETOBICOKE, ONTARIO M8W 1B3

— Fall 2011 —

We're on the web

www.tpug.ca